جامعة تشرين
كلية الهمك
قسم هندسة الإتصالات والإلكترونيات
السنة الخامسة
وظيفة 1 برمجة شبكات

First Network Programming Homework

إعداد الطالبة
الاء عمر شيخ ديب 2938
Github :

Question 1: Python Basics?

A:

If you have two lists, L1=['HTTP','HTTPS','FTP','DNS'], L2=[80,443,21,53]
convert it to generate this dictionary d={'HTTP':80,'HTTPS':443,'FTP':21,'DNS':53 }

solution:

main.py

```python
#Create and print a dictionary from two lists

#List of protocol names
L1 =["HTTP", "HTTPS", "FTP", "DNS"]
#List of port numbers
L2 =[80, 443, 21, 53]

#The zip function combines L1 and L2 into pairs like (HTTP, 80), (HTTPS, 443), etc.
#The dict function then creates a dictionary from these pairs
d = dict (zip (L1, L2))

#Print the created dictionary
print (d)
```

input

```
{'HTTP': 80, 'HTTPS': 443, 'FTP': 21, 'DNS': 53}


...Program finished with exit code 0
Press ENTER to exit console.
```

B:

Write a Python program that calculates the factorial of a given number entered by user

solution:

main.py

```python
# Calculate the factorial of a number

# Nested function to calculate factorial of a given number using recursion
# A factorial of a number n is the product of all positive integers less than or equal to n
def factorial(n):
    if n == 0:  # Base case: the factorial of 0 is defined to be 1
        return 1
    else:  # Recursive case: the factorial of n is n times the factorial of (n - 1)
        return n * factorial(n - 1)

# input() function reads a string from the user, and int() converts it to an integer
number = int(input("Enter a number to calculate its factorial: "))

# Call the factorial function with the user's number and store the result
result = factorial(number)

# Print the result of the factorial calculation
print(f"The factorial of {number} is {result}")
```

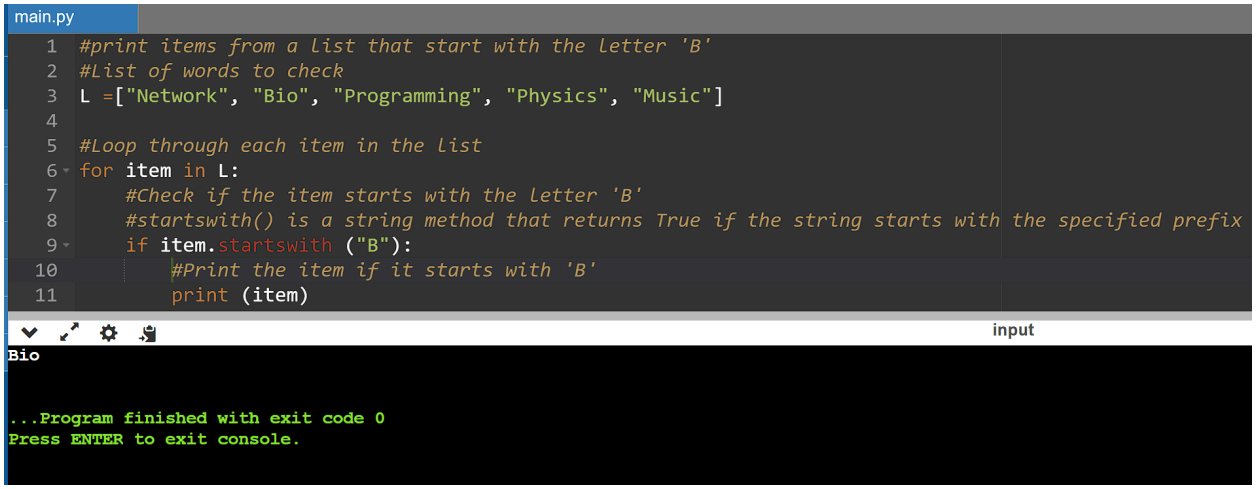
input

```
Enter a number to calculate its factorial: 4
The factorial of 4 is 24


...Program finished with exit code 0
Press ENTER to exit console.
```

C:

L=[‘Network’ , ’Bio’ , ’Programming’, ‘Physics’ , ‘Music’]
In this exercise, you will implement a Python program that reads the items of the previous list and identifies the items that starts with ‘B’ letter, then print it on screen.

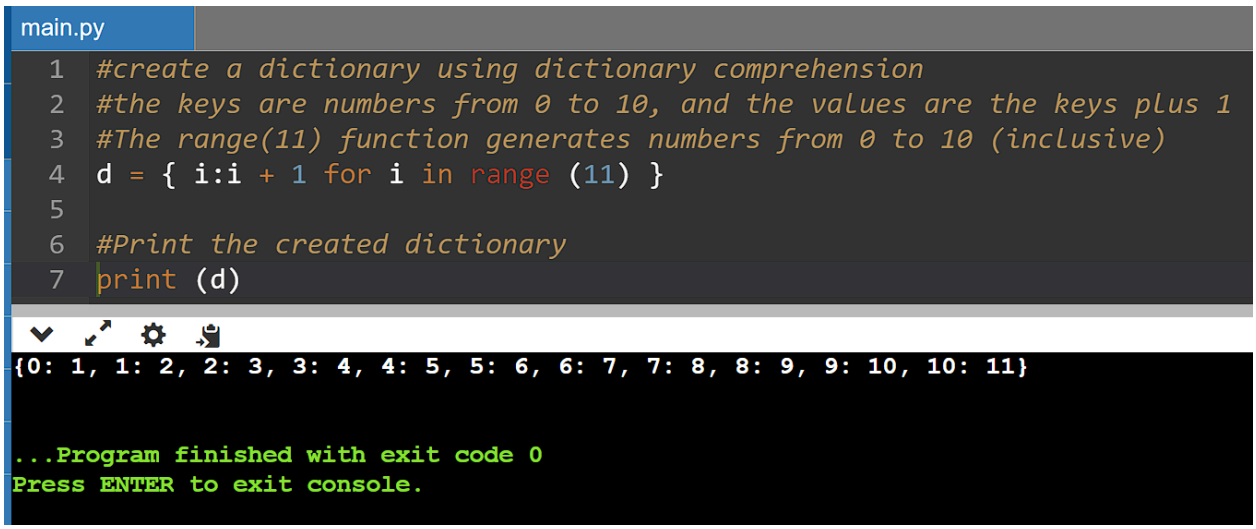
solution:



D:

Using Dictionary comprehension, Generate this dictionary
d={0:1,1:2,2:3,3:4,4:5,5:6,6:7,7:8,8:9,9:10,10:11}

solution:



Question 2: Convert from Binary to Decimal

Write a Python program that converts a Binary number into its equivalent Decimal number. The program should start reading the binary number from the user. Then the decimal equivalent number must be calculated. Finally, the program must display the equivalent decimal number on the screen.

solution:

main.py

```python
# Convert a binary string to a decimal number
def binary_to_decimal(binary_str):
    try:
        # The int function converts binary to decimal
        decimal_number = int(binary_str, 2)
        # Return the decimal number if conversion is successful
        return decimal_number
    except ValueError:
        # If there is a ValueError (invalid binary string), return None
        return None

# Repeatedly ask the user for a binary number and convert it to decimal
def start_change():
    while True:  # Start an infinite loop to keep asking the user for input
        # Prompt the user to enter a binary number
        binary_str = input("Enter a binary number: ")
        # Call the binary_to_decimal function to do the convert
        decimal_number = binary_to_decimal(binary_str)
        if decimal_number is not None:  # Check if the conversion was successful
            # Print the decimal equivalent of the binary number
            print(f"The decimal equivalent of binary {binary_str} is {decimal_number}")
            # Break the loop as a valid binary number was entered and processed
            break
        else:
            # Inform the user that the input was invalid and prompt again
            print("Invalid binary number. Please enter a valid binary number.")

# Call the start_change
start_change()
```

```
Enter a binary number: 03211
Invalid binary number. Please enter a valid binary number.
Enter a binary number: 0101011
The decimal equivalent of binary 0101011 is 43


...Program finished with exit code 0
Press ENTER to exit console.
```

Question 3: Working with Files" Quiz Program"

Type python quiz program that takes a text or json or csv file as input for (20 (Questions, Answers)). It asks the questions and finally computes and prints user results and store user name and result in separate file csv or json file.

solution:

```
import json  # Import the JSON module to handle JSON files
# load questions from a JSON file
def load_questions(filename):
    # Open the file in read mode
    with open(filename, "r") as file:
        # Load the JSON data from the file and return it
        return json.load(file)

# ask a question and check the answer
def ask_question(question_data):
    # Print the question text
    print(question_data["question"])
    # Loop through the options and print each one
    for option in question_data["options"]:
        print(option)
    # Prompt the user to choose an answer (A, B, or C)
    answer = input("Choose A, B, or C: ").strip().upper()
    # Return True if the user's answer matches the correct answer, otherwise False
    return answer == question_data["answer"]

# save quiz results to a JSON file
def save_results(filename, results):
    # Open the file in write mode
    with open(filename, "w") as file:
        # Write the results to the file in JSON format
        json.dump(results, file, indent=4)

# start the quiz
def start_quiz():
    # Load the questions from the "questions.json" file
    questions = load_questions("questions.json")
    # Initialize the count of correct answers
    correct_answers = 0
    # Get the total number of questions
```

```
    total_questions = len(questions)
    # Prompt the user to enter their name
    user_name = input("Enter your name: ")

    # Loop through each question in the list of questions
    for question in questions:
        # If the user answers the question correctly, increment the count of correct answers
        if ask_question(question):
            correct_answers += 1
    # Calculate the user's score as a percentage
    score = (correct_answers / total_questions) * 100
    # Print the user's result
    print(f"{user_name}, you have {correct_answers} correct answers out of {total_questions}" )

    # Try to load existing results from the "results.json" file
    try:
        with open("results.json", "r") as file:
            results = json.load(file)
    # If the file does not exist, create an empty list
    except FileNotFoundError:
        results = []
    # Append the user's result
    results.append({"name": user_name, "score": score})
    # Save the updated results back to the "results.json" file
    save_results("results.json", results)

# Start the quiz
start_quiz()
```

PROBLEME 26 AUSGABE DEBUGGING-KONSOLE TERMINAL PORTS KOMMENTARE

```
PS C:\Users\CS\Desktop\pythonHW1> & C:/Users/CS/AppData/Local/Microsoft
Enter your name: Alaa
What is the capital of Egypt?
A. Cairo
B. Alexandria
C. Giza
Choose A, B, or C: A
What is the capital of Saudi Arabia?
A. Jeddah
B. Mecca
C. Riyadh
Choose A, B, or C: c
What is the capital of Jordan?
A. Amman
B. Aqaba
C. Zarqa
Choose A, B, or C: a
What is the capital of Lebanon?
A. Tripoli
B. Beirut
C. Sidon
Choose A, B, or C: b
What is the capital of Iraq?
A. Basra
B. Erbil
C. Baghdad
Choose A, B, or C: c
What is the capital of Morocco?
A. Marrakech
B. Rabat
C. Casablanca
Choose A, B, or C: b
What is the capital of Tunisia?
A. Sfax
B. Sousse
C. Tunis
Choose A, B, or C: c
What is the capital of Algeria?
A. Oran
B. Algiers
C. Constantine
Choose A, B, or C: b
What is the capital of Sudan?
A. Omdurman
B. Khartoum
C. Port Sudan
Choose A, B, or C: b
What is the capital of Libya?
A. Benghazi
B. Misrata
C. Tripoli
Choose A, B, or C: c
```

PROBLEME 26 AUSGABE DEBUGGING-KONSOLE TERMINAL PORTS KOMMENTARE

```
B. Misrata
C. Tripoli
Choose A, B, or C: c
What is the capital of Yemen?
A. Aden
B. Sana'a
C. Taiz
Choose A, B, or C: b
What is the capital of Oman?
A. Muscat
B. Salalah
C. Sohar
Choose A, B, or C: a
What is the capital of Qatar?
A. Al Rayyan
B. Doha
C. Al Wakrah
Choose A, B, or C: b
What is the capital of Bahrain?
A. Manama
B. Muharraq
C. Riffa
Choose A, B, or C: a
What is the capital of Kuwait?
A. Salmiya
B. Hawalli
C. Kuwait City
Choose A, B, or C: c
What is the capital of United Arab Emirates?
A. Dubai
B. Abu Dhabi
C. Sharjah
Choose A, B, or C: a
What is the capital of Syria?
A. Aleppo
B. Homs
C. Damascus
Choose A, B, or C: c
What is the capital of Somalia?
A. Mogadishu
B. Hargeisa
C. Kismayo
Choose A, B, or C: a
What is the capital of Mauritania?
A. Nouakchott
B. Nouadhibou
C. Kiffa
Choose A, B, or C: c
What is the capital of Djibouti?
A. Tadjoura
B. Ali Sabieh
C. Djibouti City
Choose A, B, or C: c
Alaa, you have 18 correct answers out of 20
```

main.py 3 | questions.json | bank_account.py | quiz.py | results.json ●

{} results.json > ...


```json
[
    {
        "name": "asd",
        "score": 0.0
    },
    {
        "name": "Alaa ",
        "score": 15.0
    },
    {
        "name": "Alaa",
        "score": 70.0
    },
    {
        "name": "Alaa",
        "score": 0.0
    },
    {
        "name": "Alaa",
        "score": 80.0
    },
    {
        "name": "Alaa",
        "score": 90.0
    }
]
```

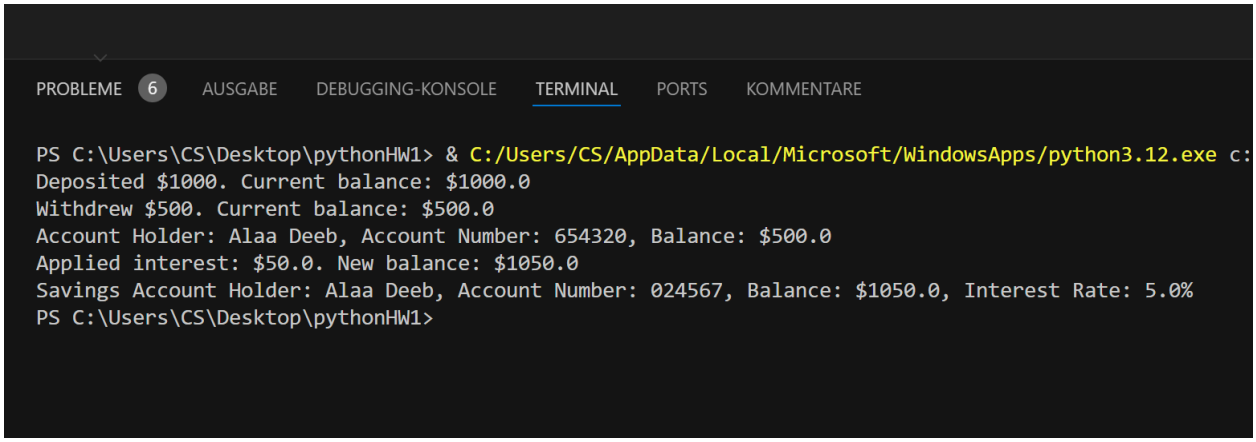
Question 4: Object-Oriented Programming - Bank Class

Define a class BankAccount with the following attributes and methods: Attributes: account_number (string), account_holder (string), balance (float, initialized to 0.0) Methods:deposit(amount), withdraw(amount) , get_balance()
- Create an instance of BankAccount, - Perform a deposit of $1000,
- Perform a withdrawal of $500. - Print the current balance after each operation.
- Define a subclass SavingsAccount that inherits from BankAccount and adds interest_rate Attribute and apply_interest() method that Applies interest to the balance based on the interest rate. And Override print() method to print the current balance and rate.
Create an instance of SavingsAccount , and call apply_interest() and print() functions.

solution:

```
# Define the BankAccount class
class BankAccount:
  # Initialize the BankAccount with account number, holder's name, and an optional balance
  def __init__(self, account_number, account_holder, balance=0.0):
    self.account_number = account_number  # Store the account number
    self.account_holder = account_holder  # Store the account holder's name
    self.balance = balance  # Store the initial balance (default is 0.0)

  # Method to deposit an amount into the account
  def deposit(self, amount):
    if amount > 0:  # Check if the deposit amount is positive
      self.balance += amount  # Add the amount to the balance
      print(
        f"Deposited ${amount}. Current balance: ${self.balance}"
      )  # Print confirmation
    else:
      print( "Deposit amount must be positive." )  # Print error if amount is not positive
  # Method to withdraw an amount from the account
  def withdraw(self, amount):
    if 0 < amount <= self.balance:  # Check if the withdrawal amount is valid
      self.balance -= amount  # Subtract the amount from the balance
      print( f"Withdrew ${amount}. Current balance: ${self.balance}" )  # Print confirmation
    else:
      print( "Insufficient balance or invalid withdrawal amount." )  # Print error if amount is invalid

  # Method to get the current balance
  def get_balance(self):
    return self.balance  # Return the current balance
```

```
  # Method to get a string representation of the account
  def __str__(self):
    return f"Account Holder: {self.account_holder}, Account Number: {self.account_number},
Balance: ${self.balance}"
# Define the SavingsAccount class, which is a subclass of BankAccount
class SavingsAccount(BankAccount):
  # Initialize the SavingsAccount with additional interest rate
  def __init__(self, account_number, account_holder, balance=0.0, interest_rate=0.01):
    super().__init__(
      account_number, account_holder, balance
    )  # Initialize the base class
    self.interest_rate = interest_rate  # Store the interest rate

  # Method to apply interest to the balance
  def apply_interest(self):
    interest = self.balance * self.interest_rate  # Calculate the interest
    self.balance += interest  # Add the interest to the balance
    print( f"Applied interest: ${interest}. New balance: ${self.balance}" )  # Print confirmation

  # Method to get a string representation of the savings account
  def __str__(self):
    return f"Savings Account Holder: {self.account_holder}, Account Number:
{self.account_number}, Balance: ${self.balance}, Interest Rate: {self.interest_rate * 100}%"

# Function to demonstrate banking operations
def start_banking():
  # Create a BankAccount instance
  account = BankAccount("654320", "Alaa Deeb")
  account.deposit(1000)  # Deposit money into the account
  account.withdraw(500)  # Withdraw money from the account
  print(account)  # Print the account details
  # Create a SavingsAccount instance
  savings_account = SavingsAccount("024567", "Alaa Deeb", balance=1000, interest_rate=0.05 )
  savings_account.apply_interest()  # Apply interest to the savings account
  print(savings_account)  # Print the savings account details
# Start the banking demonstration
start_banking()
```


PROBLEME 6
AUSGABE
DEBUGGING-KONSOLE
TERMINAL
PORTS
KOMMENTARE
PS C:\Users\CS\Desktop\pythonHW1> & C:/Users/CS/AppData/Local/Microsoft/WindowsApps/python3.12.exe c:
Deposited $1000. Current balance: $1000.0
Withdrew $500. Current balance: $500.0
Account Holder: Alaa Deeb, Account Number: 654320, Balance: $500.0
Applied interest: $50.0. New balance: $1050.0
Savings Account Holder: Alaa Deeb, Account Number: 024567, Balance: $1050.0, Interest Rate: 5.0%
PS C:\Users\CS\Desktop\pythonHW1>